Amendments to the Claims

Please replace the claims as follows:

1. (currently amended) A computer-implemented method for debugging an operating system kernel executing on a server data processing system that is coupled to a network, the kernel including a debugger control component, and the server data processing system including a network interface card that implements a debugger network component and a protocol stack, including layers from a physical layer through an application layer~~, and a debugger network component~~, comprising:

detecting debugger messages received over the network in the protocol stack;

directing the debugger messages to the debugger network component on the network interface card;

communicating the debugger messages from the debugger network component to the debugger control component in the kernel; and

performing debugging operations via the debugger control component in response to the debugger messages.

2. (original) The method of claim 1, wherein a debugger client system is coupled to the server system and further comprising:

communicating client messages from the debugger control component to the debugger network component;

directing the client messages from the debugger network component to the protocol stack; and

transmitting the client messages from the protocol stack to the client system.

3. (currently amended) The method of claim 2, wherein the kernel further includes a network interface subsystem, the method further comprising, for network messages that are not debugger messages, communicating the messages received by the protocol stack to the network interface subsystem and bypassing the debugger control component in the kernel.

4. (original) The method of claim 1, further comprising detecting the debugger messages by a port number assigned to the debugger network component.

5. (original) The method of claim 4, wherein the protocol stack implements a TCP/IP stack.

6. (original) The method of claim 1, further comprising writing the debugger messages from the debugger network component to memory of the server data processing system.

7. (original) The method of claim 1, further comprising writing the client messages from the debugger control component to memory of the server data processing system.

8. (original) An apparatus for debugging an operating system kernel executing on a server data processing system that is coupled to a network, the kernel including a debugger control component, and the server data processing system including a network interface card that implements a protocol stack, including layers from a physical layer through an application layer, and a debugger network component, comprising:

means for detecting debugger messages received over the network in the protocol stack;

means for directing the debugger messages to the debugger network component;

means for communicating the debugger messages from the debugger network component to the debugger control component in the kernel; and

means for performing debugging operations via the debugger control component in response to the debugger messages.

9. (previously presented) A computing arrangement for debugging an operating system kernel in a server system that is coupled to a client system via a network, comprising:

a memory configured in the server system;

a processor coupled to the memory and configured to execute an operating system kernel, the kernel including a debugger control component and a networking subsystem component, the debugger control component configured to perform debugging operations in response to debugger messages received over the network, and the networking subsystem configured to provide non-debugger messages to the kernel; and

a network interface circuit arrangement coupled to the processor and to the memory, the network interface circuit arrangement configured with a protocol stack and a debugger network component, the protocol stack configured to detect debugger messages received over the network and direct the debugger messages to the debugger network component, and the debugger network component configured to communicate the debugger messages to the debugger control component in the kernel.

10. (original) The arrangement of claim 9, wherein:

the debugger control component is further configured to communicate client messages from the debugger control component to the debugger network component;

the debugger network component is further configured to direct the client messages to the protocol stack; and

the protocol stack is further configured to transmit the client messages to the client system.

11. (previously presented) The arrangement of claim 10, further comprising:

a first shared memory interface coupled to the memory and to the debugger control component; and

a second shared memory interface coupled to the memory and to the debugger network component, wherein the first and second shared memory interfaces are configured to write the debugger messages and client messages to a shared memory area in the memory.

12. (original) The arrangement of claim 9, wherein the protocol stack is further configured to detect the debugger messages by a port number assigned to the debugger network component.

13. (original) The arrangement of claim 12, wherein the protocol stack is a TCP/IP stack.

14. (original) The arrangement of claim 13, wherein:

the debugger control component is further configured to communicate client messages from the debugger control component to the debugger network component;

the debugger network component is further configured to direct the client messages to the protocol stack; and

the protocol stack is further configured to transmit the client messages to the client system.

15. (previously presented) The arrangement of claim 14, further comprising:

a first shared memory interface coupled to the memory and_to the debugger control component; and

a second shared memory interface coupled to the memory and_to the debugger network component, wherein the first and second shared memory interfaces are configured to write the debugger messages and client messages to a shared memory area in the memory.

16. (previously presented) A method for debugging an operating system kernel, comprising:

executing the operating system on a server data processing system that is coupled to a network, wherein the kernel includes a debugger control component and a network interface subsystem;

identifying in a protocol stack in a network interface card, debugger messages and non-debugger messages received over the network, wherein the network interface card implements a protocol stack that includes layers form a physical layer through an application layer and a debugger network component coupled to the protocol stack;

transmitting debugger messages from the protocol stack to the debugger network component on the network interface card;

transmitting the debugger messages from the debugger network component to the debugger control component in the kernel;

transmitting non-debugger messages from the protocol stack to the network interface subsystem of the kernel; and

performing debugging operations via the debugger control component in response to the debugger messages.

17. (previously presented) The method of claim 16, wherein a debugger client system is coupled to the server system and further comprising:

transmitting client messages from the debugger control component to the debugger network component;

transmitting the client messages from the debugger network component to the protocol stack; and

transmitting the client messages from the protocol stack to the client system.

18. (previously presented) The method of claim 16, further comprising detecting the debugger messages by a port number assigned to the debugger network component.

19. (previously presented) The method of claim 18, wherein the protocol stack implements a TCP/IP stack.

20. (previously presented) The method of claim 16, further comprising:

writing the debugger messages from the debugger network component to memory of the server data processing system by a first shared memory interface in the network interface card; and

writing the client messages from the debugger control component to memory of the server data processing system by a second shared memory interface that is coupled to the debugger control component and executes on the server data processing system.

21. (previously presented) An apparatus for debugging an operating system kernel, comprising;

means for executing the operating system on a server data processing system that is coupled to a network, wherein the kernel includes a debugger control component and a network interface subsystem;

means for identifying in a protocol stack in a network interface card, debugger messages and non-debugger messages received over the network, wherein the network interface card implements a protocol stack that includes layers from a physical layer through an application layer and a debugger network component coupled to the protocol stack;

means fro transmitting debugger messages from the protocol stack to the debugger network component on the network interface card;

means for transmitting the debugger messages from the debugger network component
to the debugger control component in the kernel;

means for transmitting non-debugger messages from the protocol stack to the network interface subsystem of the kernel; and

means for performing debugging operations via the debugger control component in response to the debugger messages.